

UE INF404 - Projet Logiciel

Calculatrice : étape 2

Analyse et évaluation d'une
expression arithmétique entièrement parenthésée

L2 Informatique

Année 2022 - 2023

Rappel des précédents épisodes

Ecrire un interpréteur d'expressions arithmétiques

(~ “calculatrice en ligne”, comme la commande `bc` sous Linux)

Version initiale = “Expressions Arithmétiques Simples” (EAS)

- les opérandes sont des entiers
- opérateurs arithmétiques usuels (+, -, *, /)
- pas de priorités (évaluation de gauche à droite)

Exemples :

$25 + 2$	\rightsquigarrow	27
$25 - 4 * 2$	\rightsquigarrow	42
25	\rightsquigarrow	25
$25 + *2$	\rightsquigarrow	erreur !
-25	\rightsquigarrow	erreur !
$25 \# 2$	\rightsquigarrow	erreur !
$25/0$	\rightsquigarrow	erreur !

Rappel des précédents épisodes

Ecrire un interpréteur d'expressions arithmétiques

(~ “calculatrice en ligne”, comme la commande `bc` sous Linux)

Version initiale = “Expressions Arithmétiques Simples” (EAS)

- les opérandes sont des entiers
- opérateurs arithmétiques usuels (+, -, *, /)
- pas de priorités (évaluation de gauche à droite)

Exemples :

$25 + 2$	\rightsquigarrow	27
$25 - 4 * 2$	\rightsquigarrow	42
25	\rightsquigarrow	25
$25 + *2$	\rightsquigarrow	erreur !
-25	\rightsquigarrow	erreur !
$25 \# 2$	\rightsquigarrow	erreur !
$25/0$	\rightsquigarrow	erreur !

Rappel des précédents épisodes

Ecrire un interpréteur d'expressions arithmétiques

(~ “calculatrice en ligne”, comme la commande `bc` sous Linux)

Version initiale = “Expressions Arithmétiques Simples” (EAS)

- les opérandes sont des entiers
- opérateurs arithmétiques usuels (+, -, *, /)
- pas de priorités (évaluation de gauche à droite)

Exemples :

$25 + 2$	\rightsquigarrow	27
$25 - 4 * 2$	\rightsquigarrow	42
25	\rightsquigarrow	25
$25 + *2$	\rightsquigarrow	erreur !
-25	\rightsquigarrow	erreur !
$25 \# 2$	\rightsquigarrow	erreur !
$25/0$	\rightsquigarrow	erreur !

Langage des EAS

① Alphabet :

$V = \{0, 1, 2, \dots, 9, +, -, *, /, \text{espace}, \text{tabulation}, \text{fin-de-ligne}\}$

② Lexique :

Deux classes de lexèmes, expressions régulières (ou automates) sur V

- ▶ ENTIER = chiffre.(chiffre)* et
- ▶ OPERATEUR = '+' + '-' + '*' + '/'

③ Syntaxe :

- ▶ séquence alternée d'ENTIER et OPERATEUR
- ▶ débutant et finissant par ENTIER

→ expression régulière (ou automate) sur les lexèmes :

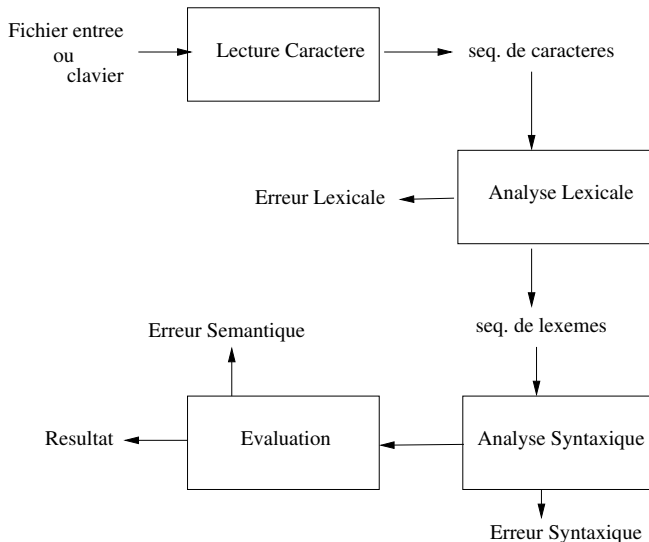
$\text{entier} \cdot (\text{opérateur} \cdot \text{entier})^*$

④ Sémantique :

pas de division par 0 ...

Structure de la calculatrice

Quatre composants/modules principaux ...



La suite ?

Généralisation : priorités, donc parenthèses ...

ex : $5 + 3 * 4 = 17$ $(5 + 3) * 4 = 32$

Nouveau langage d'entrée :

- 1 étendre l'alphabet : nouveaux caractères '(' et ')'
- 2 étendre le lexique : nouveaux lexèmes : PARO et PARF
- 3 étendre la syntaxe :
règle de "bonne imbrication des parenthèses"
- 4 modifier la sémantique :
 - ▶ toujours pas de division par 0 ...
 - ▶ mais nouvelles règles de calcul (priorité des opérateurs)

Conséquences en terme de programmation ?

- 1 analyse lexicale ?
- 2 analyse syntaxique ?
- 3 évaluation ?

La suite ?

Généralisation : priorités, donc parenthèses ...

ex : $5 + 3 * 4 = 17$ $(5 + 3) * 4 = 32$

Nouveau langage d'entrée :

- ❶ étendre l'alphabet : nouveaux caractères ' (' et ') '
- ❷ étendre le lexique : nouveaux lexèmes : PARO et PARF
- ❸ étendre la syntaxe :
règle de "bonne imbrication des parenthèses"
- ❹ modifier la sémantique :
 - ▶ toujours pas de division par 0 ...
 - ▶ mais nouvelles règles de calcul (priorité des opérateurs)

Conséquences en terme de programmation ?

- ❶ analyse lexicale ?
- ❷ analyse syntaxique ?
- ❸ évaluation ?

La suite ?

Généralisation : priorités, donc parenthèses ...

$$\text{ex : } 5 + 3 * 4 = 17 \quad (5 + 3) * 4 = 32$$

Nouveau langage d'entrée :

- ❶ étendre l'alphabet : nouveaux caractères ' (' et ') '
- ❷ étendre le lexique : nouveaux lexèmes : PARO et PARF
- ❸ étendre la syntaxe :
règle de "bonne imbrication des parenthèses"
- ❹ modifier la sémantique :
 - ▶ toujours pas de division par 0 ...
 - ▶ mais nouvelles règles de calcul (priorité des opérateurs)

Conséquences en terme de programmation ?

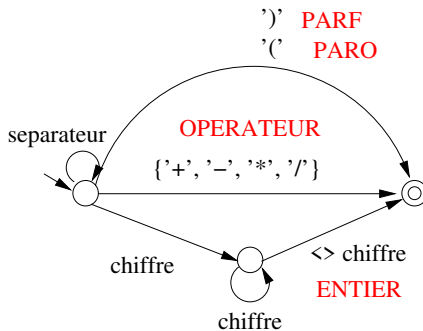
- ❶ analyse lexicale ?
- ❷ analyse syntaxique ?
- ❸ évaluation ?

Analyse Lexicale

Deux nouvelles classes de lexèmes :

- PARO = '('
- PARF = ')'

⇒ Etendre l'automate de `reconnaitre_lexeme()` :



Analyse syntaxique et évaluation

Spécifier la syntaxe d'une expression arithmétique "parenthésée" ?

Utiliser un **automate** ?

- compter les parenthèses ouvrantes et fermantes (même nombre !)
- vérifier leur bonne imbrication :

$(5 * (2 + 3)) \rightsquigarrow \text{OK}$

$(5 * 2)) + (3 \rightsquigarrow \text{Erreur}$

→ impossible avec un automate à nombre **fini** d'états !

c.f. INF302, langages **régulier**, $a^n.b^n$ **non régulier** ...

⇒ définir un **nouveau formalisme** ?

Evaluation ?

$(5 * (2 * (3 + 1)))$

- quand/comment faire les calculs ?
- mémoriser un préfixe de l'expression lue ? ?

Analyse syntaxique et évaluation

Spécifier la syntaxe d'une expression arithmétique "parenthésée" ?

Utiliser un **automate** ?

- compter les parenthèses ouvrantes et fermantes (même nombre !)
- vérifier leur bonne imbrication :

$(5 * (2 + 3)) \rightsquigarrow \text{OK}$

$(5 * 2)) + (3 \rightsquigarrow \text{Erreur}$

→ impossible avec un automate à nombre **fini** d'états !

c.f. INF302, langages **régulier**, $a^n.b^n$ **non régulier** ...

⇒ **définir un nouveau formalisme ?**

Evaluation ?

$(5 * (2 * (3 + 1)))$

- quand/comment faire les calculs ?
- mémoriser un préfixe de l'expression lue ? ?

Analyse syntaxique et évaluation

Spécifier la syntaxe d'une expression arithmétique "parenthésée" ?

Utiliser un **automate** ?

- compter les parenthèses ouvrantes et fermantes (même nombre !)
- vérifier leur bonne imbrication :

$(5 * (2 + 3)) \rightsquigarrow \text{OK}$

$(5 * 2)) + (3 \rightsquigarrow \text{Erreur}$

→ impossible avec un automate à nombre **fini** d'états !

c.f. INF302, langages **régulier**, $a^n.b^n$ **non régulier** ...

⇒ **définir un nouveau formalisme ?**

Evaluation ?

$(5 * (2 * (3 + 1)))$

- quand/comment faire les calculs ?
- mémoriser un préfixe de l'expression lue ? ?

Etape intermédiaire : les EAEP

EAEP = Expressions Arithmétiques Entièrement Parenthésées

Idée = écrire **toute opération** entre parenthèses ...

Exemples :

25	\rightsquigarrow	25
(25 + 2)	\rightsquigarrow	27
((25 + 2))	\rightsquigarrow	27
((25 - 4) * 2)	\rightsquigarrow	42
(25 - (4 * 2))	\rightsquigarrow	17
(25 \$ 2)	\rightsquigarrow	erreur! (<i>lexicale</i>)
25 + *2	\rightsquigarrow	erreur! (<i>syntactique</i>)
25 + 4 * 2)	\rightsquigarrow	erreur! (<i>syntactique</i>)
25 + (4 * 2)	\rightsquigarrow	erreur! (<i>syntactique</i>)
(25/(5 - 5))	\rightsquigarrow	erreur! (<i>sémantique</i>)

Etape intermédiaire : les EAEP

EAEP = Expressions Arithmétiques Entièrement Parenthésées

Idée = écrire **toute opération** entre parenthèses ...

Exemples :

25	\rightsquigarrow	25
(25 + 2)	\rightsquigarrow	27
((25 + 2))	\rightsquigarrow	27
((25 - 4) * 2)	\rightsquigarrow	42
(25 - (4 * 2))	\rightsquigarrow	17
(25 \$ 2)	\rightsquigarrow	erreur! (<i>lexicale</i>)
25 + *2	\rightsquigarrow	erreur! (<i>syntactique</i>)
25 + 4 * 2)	\rightsquigarrow	erreur! (<i>syntactique</i>)
25 + (4 * 2)	\rightsquigarrow	erreur! (<i>syntactique</i>)
(25/(5 - 5))	\rightsquigarrow	erreur! (<i>sémantique</i>)

Définir une eaep ?

L'ensemble eaep est défini par les règles suivantes :

- 1 ENTIER est une eaep
- 2 PARO eaep PLUS eaep PARF est une eaep
- 3 PARO eaep MOINS eaep PARF est une eaep
- 4 PARO eaep MUL eaep PARF est une eaep
- 5 PARO eaep DIV eaep PARF est une eaep

→ définition **récursive** des eaep
(schéma inductif, ~ schéma de récurrence)

Définir une eaep ?

L'ensemble eaep est défini par les règles suivantes :

- 1 ENTIER est une eaep
- 2 PARO eaep PLUS eaep PARF est une eaep
- 3 PARO eaep MOINS eaep PARF est une eaep
- 4 PARO eaep MUL eaep PARF est une eaep
- 5 PARO eaep DIV eaep PARF est une eaep

→ définition **récursive** des eaep
(schéma inductif, ~ schéma de récurrence)

Définir une eaep ?

L'ensemble eaep est défini par les règles suivantes :

- 1 ENTIER est une eaep
- 2 PARO eaep PLUS eaep PARF est une eaep
- 3 PARO eaep MOINS eaep PARF est une eaep
- 4 PARO eaep MUL eaep PARF est une eaep
- 5 PARO eaep DIV eaep PARF est une eaep

→ définition **récursive** des eaep
(schéma inductif, ~ schéma de récurrence)

Définir une eaep ?

L'ensemble eaep est défini par les règles suivantes :

- ① ENTIER est une eaep
- ② PARO eaep PLUS eaep PARF est une eaep
- ③ PARO eaep MOINS eaep PARF est une eaep
- ④ PARO eaep MUL eaep PARF est une eaep
- ⑤ PARO eaep DIV eaep PARF est une eaep

→ définition **récursive** des eaep
(schéma inductif, \sim schéma de récurrence)

Autre écriture : grammaire des eaep

$exp \rightarrow eaep \text{ FIN_SEQUENCE}$
 $eaep \rightarrow \text{ENTIER}$
 $eaep \rightarrow \text{PARO } eaep \text{ op } eaep \text{ PARF}$
 $op \rightarrow \text{PLUS}$
 $op \rightarrow \text{MOINS}$
 $op \rightarrow \text{MUL}$

- $\{\text{FIN_SEQUENCE}, \text{ENTIER}, \text{PARO}, \text{PARF}, \text{PLUS}, \text{etc}\} =$ lexèmes du langage = vocabulaire **terminal** de la grammaire
- $\{exp, op, eaep\} =$ vocabulaire **non-terminal** de la grammaire
- $exp =$ **axiome** de la grammaire

Plus d'infos sur les grammaires au cours suivant !

Exemple de reconnaissance d'une EAEP (1)

Reconnaitre 25 ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ ENTIER FIN_SEQUENCE}$

Reconnaitre (25 + 2) ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS ENTIER PARF FIN_SEQUENCE}$

Exemple de reconnaissance d'une EAEP (1)

Reconnaitre 25 ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ ENTIER FIN_SEQUENCE}$

Reconnaitre (25 + 2) ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS ENTIER PARF FIN_SEQUENCE}$

Exemple de reconnaissance d'une EAEP (1)

Reconnaitre 25 ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ ENTIER FIN_SEQUENCE}$

Reconnaitre (25 + 2) ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS ENTIER PARF FIN_SEQUENCE}$

Exemple de reconnaissance d'une EAEP (1)

Reconnaitre 25 ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ ENTIER FIN_SEQUENCE}$

Reconnaitre (25 + 2) ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS ENTIER PARF FIN_SEQUENCE}$

Exemple de reconnaissance d'une EAEP (1)

Reconnaitre 25 ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ ENTIER FIN_SEQUENCE}$

Reconnaitre (25 + 2) ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS ENTIER PARF FIN_SEQUENCE}$

Exemple de reconnaissance d'une EAEP (1)

Reconnaitre 25 ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ ENTIER FIN_SEQUENCE}$

Reconnaitre (25 + 2) ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS ENTIER PARF FIN_SEQUENCE}$

Exemple de reconnaissance d'une EAEP (1)

Reconnaitre 25 ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ ENTIER FIN_SEQUENCE}$

Reconnaitre (25 + 2) ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS ENTIER PARF FIN_SEQUENCE}$

Exemple de reconnaissance d'une EAEP (1)

Reconnaitre 25 ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ ENTIER FIN_SEQUENCE}$

Reconnaitre (25 + 2) ?

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS ENTIER PARF FIN_SEQUENCE}$

Exemple de reconnaissance d'une EAEP (1)

Reconnaitre 25 ?

$$\begin{aligned} \text{exp} &\Rightarrow_G \text{eaep FIN_SEQUENCE} \\ &\Rightarrow_G \text{ENTIER FIN_SEQUENCE} \end{aligned}$$

Reconnaitre (25 + 2) ?

$$\begin{aligned} \text{exp} &\Rightarrow_G \text{eaep FIN_SEQUENCE} \\ &\Rightarrow_G \text{PARO eaep op eaep PARF FIN_SEQUENCE} \\ &\Rightarrow_G \text{PARO ENTIER op eaep PARF FIN_SEQUENCE} \\ &\Rightarrow_G \text{PARO ENTIER PLUS eaep PARF FIN_SEQUENCE} \\ &\Rightarrow_G \text{PARO ENTIER PLUS ENTIER PARF FIN_SEQUENCE} \end{aligned}$$

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G PARO \text{ } eaep \text{ } op \text{ } eaep \text{ } PARF \text{ FIN_SEQUENCE}$
 $\Rightarrow_G PARO \text{ ENTIER } op \text{ } eaep \text{ } PARF \text{ FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G PARO \text{ } eaep \text{ } op \text{ } eaep \text{ } PARF \text{ FIN_SEQUENCE}$
 $\Rightarrow_G PARO \text{ ENTIER } op \text{ } eaep \text{ } PARF \text{ FIN_SEQUENCE}$
 $\Rightarrow_G PARO \text{ ENTIER PLUS } eaep \text{ } PARF \text{ FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER } op \text{ eaep } \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER } op \text{ eaep } \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER } op \text{ eaep } \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER } op \text{ eaep } \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Exemple de reconnaissance d'une EAEP (2)

Reconnaitre (25 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : opérateur manquant

Reconnaitre (25 + 2)

$exp \Rightarrow_G eaep \text{ FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO } eaep \text{ op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER op } eaep \text{ PARF FIN_SEQUENCE}$
 $\Rightarrow_G \text{ PARO ENTIER PLUS } eaep \text{ PARF FIN_SEQUENCE}$

\Rightarrow Erreur : parenthèse fermante manquante

Vérifier la syntaxe d'une eaep

→ un algo récursif, **basé sur la grammaire**

eaep → ENTIER

eaep → PARO *eaep* op *eaep* PARF

Rec_eaep =

selon LC().nature // LC est le lexeme_courant()

cas ENTIER : Avancer

cas PARO : Avancer ; Rec_eaep ; Rec_op ; Rec_eaep ;

si LC().nature = PARF alors Avancer sinon Erreur

autre : Erreur

fin

op → PLUS

op → MOINS

op → MUL

Rec_op =

selon LC().nature

cas PLUS, MUL, MOINS : Avancer

autre : Erreur

fin

Vérifier la syntaxe d'une eaep

→ un algo récursif, **basé sur la grammaire**

eaep → ENTIER

eaep → PARO *eaep* *op* *eaep* PARF

```
Rec_eaep =  
  selon LC().nature // LC est le lexeme_courant()  
    cas ENTIER : Avancer  
    cas PARO : Avancer ; Rec_eaep ; Rec_op ; Rec_eaep ;  
      si LC().nature = PARF alors Avancer sinon Erreur  
    autre : Erreur  
fin
```

op → PLUS

op → MOINS

op → MUL

```
Rec_op =  
  selon LC().nature  
    cas PLUS, MUL, MOINS : Avancer  
    autre : Erreur  
fin
```

Evaluer une eaep

Complétez l'analyse syntaxique ...

```
// on ajoute un paramètre résultat qui renvoie la valeur de l'expression
Rec_eaep (resultat valeur : entier) =
  selon LC().nature
    cas ENTIER : valeur = LC().valeur ; Avancer
    cas PARO : Avancer ;
                  Rec_eaep (valg) ; Rec_op (op) ; Rec_eaep (vald) ;
                  valeur = Evaluer (valg, op, vald)
    si LC().nature = PARF alors Avancer sinon Erreur
    autre : Erreur
fin
```

Avec :

- Rec_op(resulat op : operateur);
reconnait un opérateur et l'affecte au paramètre op
- Evaluer (vald : entier, op : operateur, valg : entier): un entier
renvoie la valeur de vald op valg

Le TP2

Programmer une version de la calculette pour les EAEPs ...

- ① analyse lexicale : ajouter PARO et PARF
 - ▶ étendre le type `Lexeme`
 - ▶ étendre `reconnaitre_lexeme()`, `afficher_lexeme`, etc
 - ▶ tester avec `test_lexeme` !

- ② analyse syntaxique :
 - ▶ écrire la procédure `analyser()`, basée sur la grammaire (appel à `Rec_eaep()` ...)
 - ▶ tester !

- ③ évaluation :
 - ▶ compléter la procédure `analyser()` ...
 - ▶ tester !

Conclusion (provisoire !)

- un formalisme pour décrire la syntaxe d'un langage non régulier
→ les **grammaires**
- on peut **déduire** de ce formalisme
un algorithme **systématique** d'**analyse syntaxique**
(toujours ???)

⇒ Mise en oeuvre au TP2, et généralisation aux cours suivants !